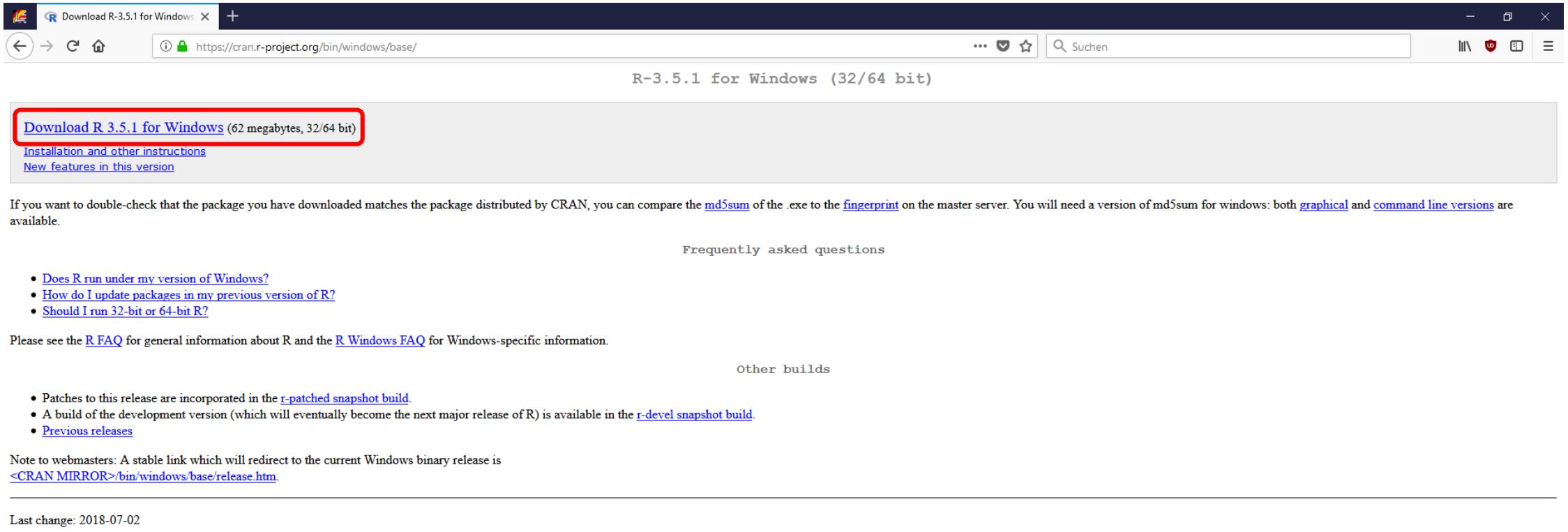
1) Download R

- This plugin requires R to be installed and configured on your computer
- All steps in this readme have been tested on Windows computers only
- For Windows, go to https://cran.r-project.org/bin/windows/base/
- Download and install „R X.X.X. for Windows“
- *Important: during the installation, keep the standard settings, especially for installation path and registry settings*

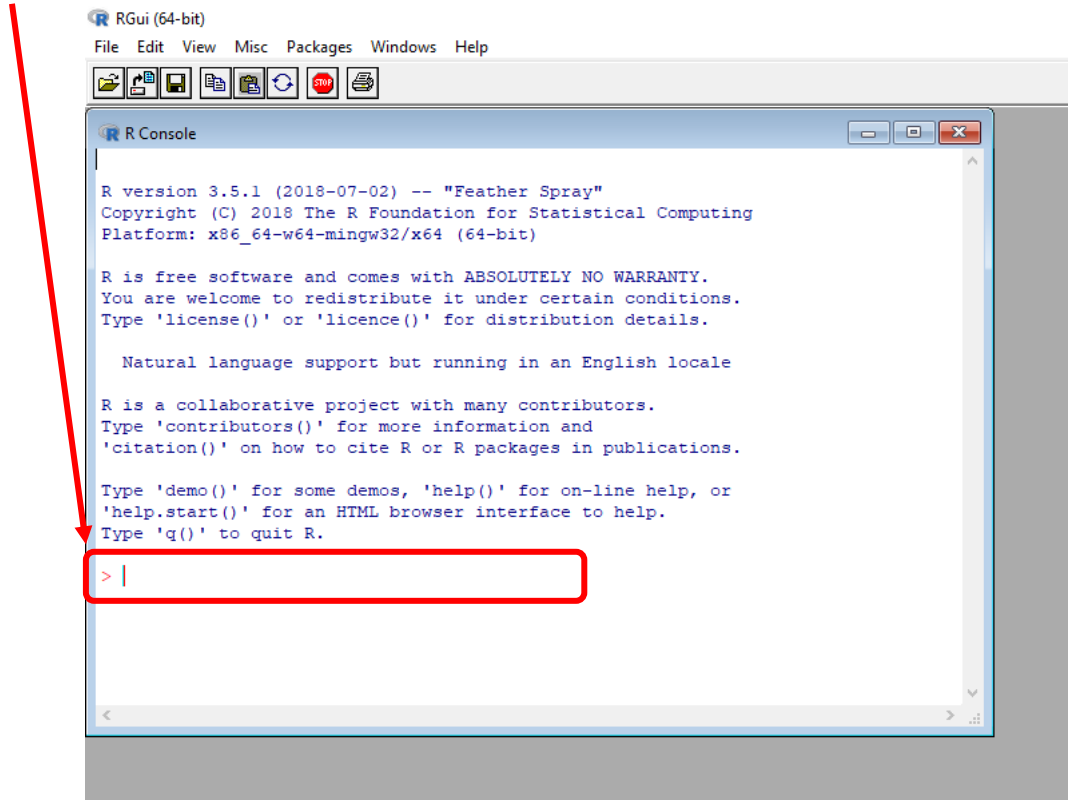
IMPORTANT: If you already have R installed, make sure it is up-to-date (at least version 3.5.1)!
If it is older, uninstall R FIRST, then install the newest version of R.



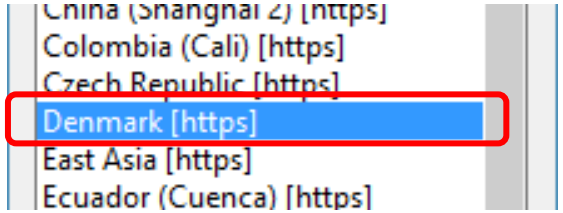
2) Prepare R to work with Perseus

- Start the R x64bit version (e.g. „R x64 X.X.X“ in the start menu)
- Copy the following 4 code lines *individually* and paste them into the console window, confirm each with enter

install.packages('BiocManager')
BiocManager::install('Biobase', ask = F)
install.packages('PerseusR')
library(PerseusR)



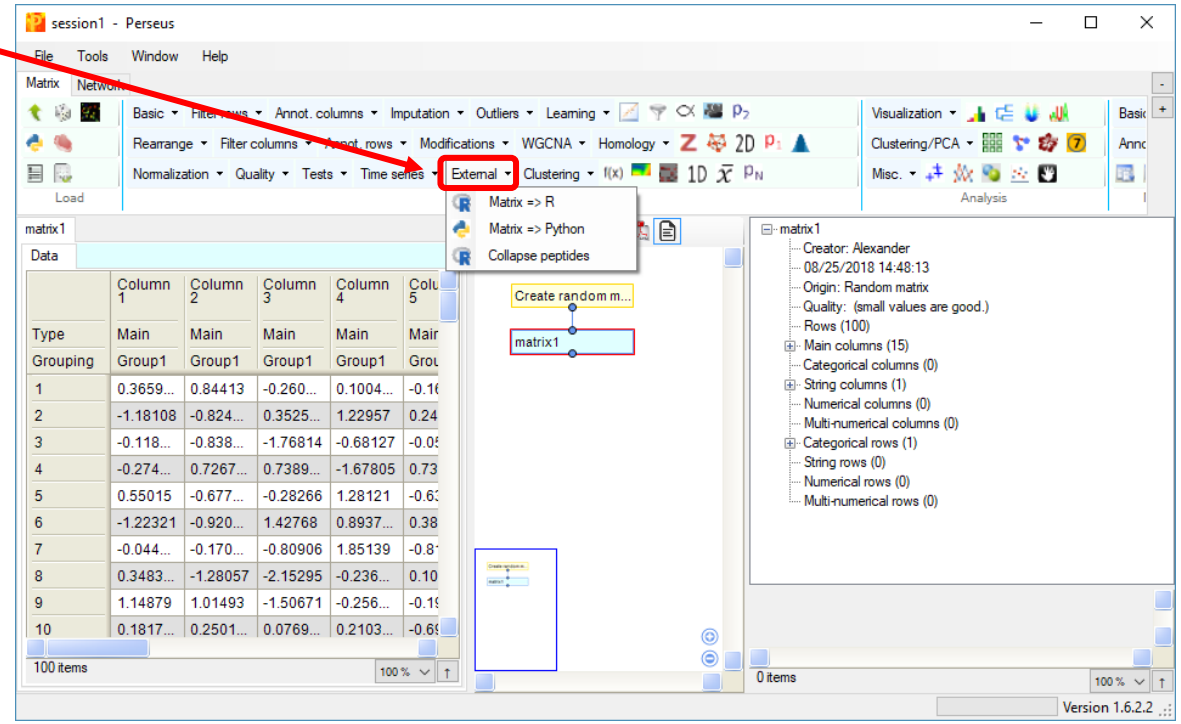
- Windows 10: there might be an error message „library writing failed“ followed by the question if you want a personal library instead
  - Accept with yes, and again yes on the suggested directory
- You will be asked to pick a CRAN mirror: pick the closest one

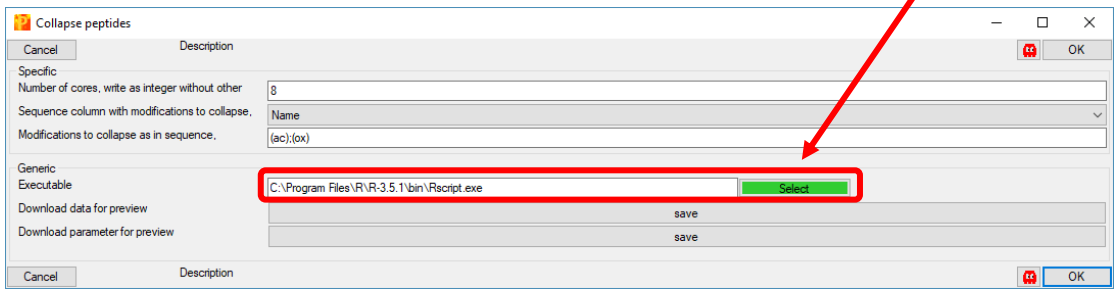


- Code lines 1 to 3 should result in R downloading data
- You can now close R (don‘t save the workspace image)
- If it worked, you will (probably ☺) not have to repeat these steps ever again

3) Run Plugins within Perseus

- Make sure you use at least Perseus v1.6.2 for the plugin to work! I tested it with v1.6.2.2
- To run a plugin from within Perseus, the plugin file („PluginPeptideCollapse.dll“) must be placed within the Perseus/bin/ folder
  - *Do not rename the plugin file, otherwise it might not work anymore*
  - *If copying produces an error, make sure that Perseus is closed and try again*
- After copying the plugin into the „bin“ folder, Perseus must be restarted if it was already opened
- Within Perseus, plugins are collected in the folder „External“
- Some plugins do not require R and work like all other tools in Perseus
- But a plugin based on R, such as „Collapse peptides“, needs to know the R program path
- If slide 2) was done correctly, the field „Exectuable“ is already filled in, and the button „Select“ is highlighted green
- If this is not the case, there was an error in slide 2) -> try executing the code lines again



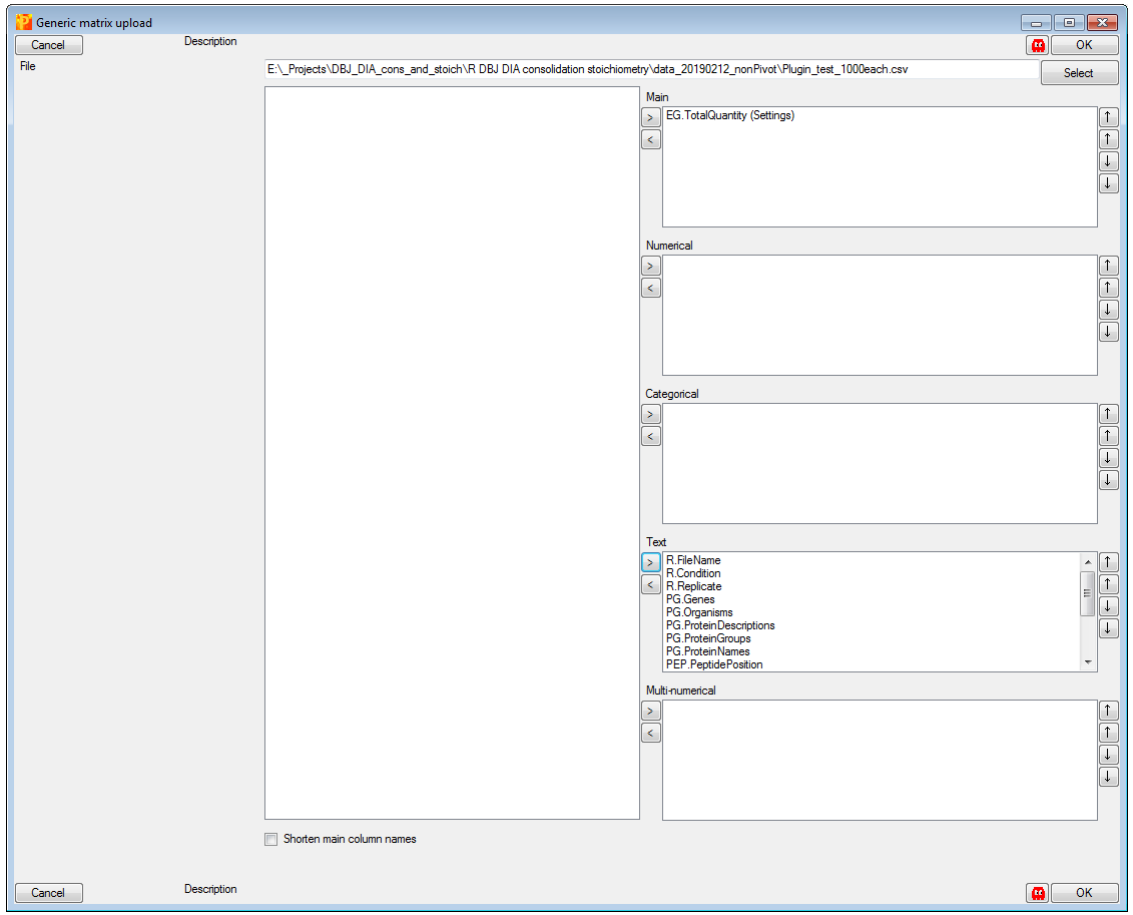


Readme PeptideCollapse 1) Load Spectronaut long-format (not Pivot!) into Perseus

Important: have to rename ".xls" Spectronaut report to ".txt" so that Perseus can find it

- Alternatively: copy the ".xls" file name into the Perseus import dialog -> it will load even though the file is not shown

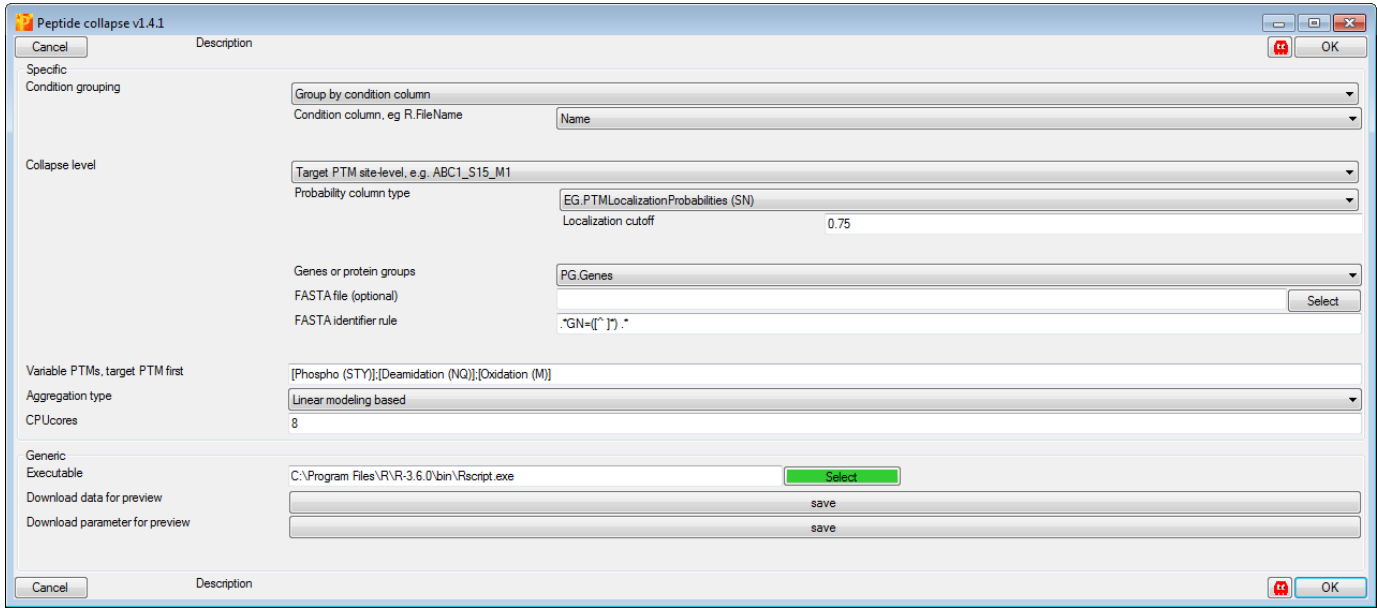
- Intensity column (e.g. "EG.TotalQuantity (Settings)") has to go into main
- All other columns go into txt (see next slide for which columns are needed for the plugin to work!!!)



Readme PeptideCollapse 2) Set up the plugin

If the plugin was installed correctly as described on slides 1-3, it should show up as “Peptide Collapse v1.X” under “External”. Several options can be set:

- Format input and condition renaming: chose the column describing experimental conditions (e.g. EGF_01, EGF_02, Control_01, Control_02,…), e.g. R.FileName or R.Condition
  - This column will be used to transform the long-table report into a wide-table report
  - Chose “skip” if your data is already in wide-table format
- Collapse level: chose if peptide precursors should be collapsed on site-, target PTM peptide- or modification-specific peptide-level
  - Probability column type: chose if localizations are taken from **EG.PTMLocalizationProbabilities**, **EG.PTMAssayProbability** or ignored
  - Selecting EG.PTMAssayProbability will perform peptide precursor grouping based on **EG.PrecursorId**
  - Regardless of which column is used for filtering, **EG.PrecursorId** needs to be provided in addition to read out charge state information
  - Localization cutoff: if this value is set to 0, nothing happens; otherwise will kick out all peptides with localization probability below the set cutoff (e.g. 0.75 or 0.99)
  - Site-level collapse requires **PEP.PeptidePosition** and either **PG.Genes** or **PG.ProteinGroups** to be provided as text columns
  - Site-level collapse optionally allows PTM sequence motif extraction if a **FASTA file** is provided; a parsing rule for the header (e.g. “.*GN=([^]*) .*“ for genes or “.*\|(.*)\|.*” for proteins; depends on the FASTA file used) needs to be provided
  - Target PTM peptide-level allows stoichiometry calculation also for MaxQuant evidence files
    - In this case columns “**Modified Sequence**” and a target probability column (e.g. “**Phospho (STY) Probabilities**”) need to be provided
- Variable PTMs: **list here ALL the PTMs** included in either EG.PTMLocalizationProbabilities or EG.PrecursorId (for filtering on EG.PTMAssayProbability), separated by “;”
  - Site-level collapse and target PTM peptide-level stoichiometry calculation are performed for the first PTM listed here
  - If a MaxQuant evidence file is used, list instead ALL the PTMs included in the “Modified sequence” column, e.g. (ph);(ac);(ox)
- Aggregation type: chose if missing values between collapsed peptides should be extrapolated via linear regression, or simply summed
- CPUcores: set the number of threads you want to use



Readme PeptideCollapse 2) Results

When the plugin is done, a new matrix is displayed in Perseus -> it shows the intensity conditions as main columns, and all other columns as text columns. To process them, you can either:

- Save the data outside of Perseus and re-import it manually, which allows you to re-load columns as numeric,…
- Manually reassign column types in Perseus

New columns:

- PTM_0_num: lists the number of target PTMs on peptide-level
- PTM_group: lists the peptide precursors that were collapsed, separated by ";"
- PTM_collapse_key: used for site-level collapse; lists the gene/protein identifier _ PTM amino acid type & position _ multiplicity (equals PTM_0_num, but capped at 3 max)
- PTM_collapse_key_entries: lists how many peptide precursors were collapsed into the key sequence
- PTM_localization (if localization filtering selected): this column is the maximum observed localization probability for each site

New columns for site-level collapse:

- PTM_seq (if FASTA file provided): lists 31 aa PTM sequence motif around target PTM

New columns for target PTM peptide-level collapse:

- Occ_[condition names] (if stoichiometry calculation selected): these columns list stoichiometry information (= occupancies) for each condition
- PTM_stoich_key (if stoichiometry calculation selected): this column lists the peptide base sequence used to calculated stoichiometry values
- PTM_stoich_key_PTMnum (if stoichiometry calculation selected): this column lists the number of target PTMs of all peptides used for one PTM_stoich_key model

Readme PeptideCollapse 3) Example File

The file “Plugin_peptide_collapse_test.csv” is an example dataset, which can be used to collapse data. (It will not yield stoichiometry, since it is a phospho-only dataset)
It is based on the Spectronaut v13 standard report, which is in long-table format (= there is only 1 intensity column and different conditions are reported in an extra column).
To run the plugin with it, follow these steps:

- Import the file via “generic matrix upload” into Perseus (set the file type to “.csv” so that you can see it)
- Load “EG.TotalQuantity (Settings)” as a main column, and all other columns as text columns
- Load the plugin via External -> Peptide collapse and adjust the settings:
    - The data is in long-table format, so the plugin needs a condition column to group it -> select R.Condition
    - Select which level to collapse into: target PTM site-level, target PTM peptide-level or ModSpec peptide-level (imitating the MaxQuant modification specific peptide format)
    - In this example, we will select “target PTM site-level”
        - We select “EG.PTMLocalizationProbabilities (SN)” and set the localization cutoff to 0.75
        - We select PG.Genes (= sites will be collapsed on gene level) and load a “HUMAN.fasta” Uniprot FASTA file to create sequence windows
    - For the variable PTMs, we need to set phospho first, since this will define the site-level for the collapse
        - We also searched the data with deamidation and oxidation as variable PTMs
        - We thus write “[Phospho (STY)];[Deamidation (NQ)];[Oxidation (M)]”
    - We leave the other settings as standard and execute the plugin
- After some time, Perseus should show a new matrix with our site-level collapsed data. Now, there are 18 main columns (= conditions) and 923 rows (= phospho-sites).